Deep Neural Net Approaches for Natural Language Processing

Gary Geunbae Lee
Dept of CSE/AIGS
POSTECH
Deep Learning Basics
AI vs. ML vs. DL

AI: Artificial Intelligence
ML: Machine Learning
DL: Deep Learning

Artificial Intelligence
- Reasoning
- Knowledge Representation
- Planning
- Learning
- Perception
- Manipulation

Machine Learning
- Supervised Learning with Teachers
- Semi-supervised Learning with Teacher
- Unsupervised Learning without Teacher
- Reinforcement Learning with Rewards

Deep Learning
- RBM (Restricted Boltzmann Machine)
- DBN (Deep Belief Network)
- CNN (Convolutional Neural Network)
- Deep Reinforcement Learning

Input
- Text
- Sound
- Image
- Video

Output
- Generated Information
- Autonomous Control
Artificial Neural Networks (ANN)
Layered Networks

Output: \[ y = f(w^1 x + w^2 x + w^3 x + \cdots + w^m x_m) \]

\[ = f(\sum_j w^j x_j) \]
Deep learning Innovation

- Combining Feature Learning and Classification as Unified Framework (※ Learning what to learn, how to learn)
Vanilla recurrent neural networks (RNNs)

• RNNs have connections from the outputs of previous time steps to inputs of next time steps

• For sequential data, a RNN usually computes hidden state $h_t$ from the previous hidden state $h_{t-1}$ and the input $x_t$
  • $h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$
Vanishing gradient problem

- \( h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \)
- Let’s assume \( \sigma \) is the identity function

\[
\frac{\partial J^{(i)}(\theta)}{\partial h^{(j)}} = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} \frac{\partial h^{(t)}}{\partial h^{(t-1)}}
\]

\[
= \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} W_h = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} W^\ell_h
\]

If all \( \frac{\partial h_t}{\partial h_{t-1}} < 1 \) \( \Rightarrow \frac{\partial J^t}{\partial h^1} \approx 0 \)
Long short-term memory networks (LSTMs)

- LSTMs explicitly keep and update cell memory $c(t)$ by
  - Removing the previous cell content $c(t-1)$ by multiplying it with $f(t)$
  - Adding the new cell content $\tilde{c}(t)$ multiplied by $i(t)$
- LSTMs produce output $h(t) = o(t) \circ \tanh c(t)$

\[
\begin{align*}
  f(t) &= \sigma \left( W_f h(t-1) + U_f x(t) + b_f \right) \\
  i(t) &= \sigma \left( W_i h(t-1) + U_i x(t) + b_i \right) \\
  o(t) &= \sigma \left( W_o h(t-1) + U_o x(t) + b_o \right) \\
  \tilde{c}(t) &= \tanh \left( W_c h(t-1) + U_c x(t) + b_c \right) \\
  c(t) &= f(t) \circ c(t-1) + i(t) \circ \tilde{c}(t) \\
  h(t) &= o(t) \circ \tanh c(t)
\end{align*}
\]
Gated recurrent units (GRUs)

- GRUs keeps and update $h^{(t)}$ by two gates:
  - Update gate $u^{(t)}$ decides
    - How much the old hidden representation $h^{(t)}$ is removed
    - How much the new hidden representation $\tilde{h}^{(t)}$ is added
  - Reset gate $r^{(t)}$ decides how much old representation $h^{(t)}$ is needed to compute new representation $\tilde{h}^{(t)}$

- GRUs also use less number of gates and have smaller parameters than LSTMs

$$
\begin{align*}
    u^{(t)} &= \sigma \left( W_u h^{(t-1)} + U_u x^{(t)} + b_u \right) \\
    r^{(t)} &= \sigma \left( W_r h^{(t-1)} + U_r x^{(t)} + b_r \right) \\
    \tilde{h}^{(t)} &= \tanh \left( W_h ( r^{(t)} \circ h^{(t-1)} ) + U_h x^{(t)} + b_h \right) \\
    h^{(t)} &= (1 - u^{(t)}) \circ h^{(t-1)} + u^{(t)} \circ \tilde{h}^{(t)}
\end{align*}
$$
Bidirectional Multi-Layer RNNs

the  movie  was  terribly  exciting  !
Parallel computing for Deep Learning

- History of parallel/distributed systems for Deep Learning computing

- Google taps 16k computers to look for cats—for Science!
- Univ. of Toronto uses 2 GPUs for 1.2M training images for 1000 classes image classification (ImageNet Large Scale Visual Recognition Challenge)
- Stanford uses 12 GPUs for Large-scale Video Classification With Convolutional Neural Networks (10M Youtube video)
- Google uses 16K CPU cores for Training 22-layers Deep neural network (GoogLeNet, 2014)
- Baidu’s Artificial Intelligence Supercomputer Beats Google at Image Recognition
Deep Learning for NLP
Word Vector

- Represent words as vectors

\[ \text{expect} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix} \]
Word Vector

- Distributional semantics: A word’s meaning is given by the words that frequently appear close-by.
- "You shall know a word by the company it keeps"
- Word2vec objective function (skip-grams)

\[ J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} | w_t; \theta) \]
Contextual word embedding

• A word’s **contextual embedding** must consider its context

\[
\epsilon(\text{plays})
\]

\[
\epsilon(\text{plays} \mid \text{the actor \_ a show})
\]

GloVe

the actor plays a show

Some contextual method

the actor plays a show
ELMo: Embeddings from Language Model

- Multi-layer bidirectional LSTM language model

\[
\begin{align*}
R_k &= \{x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} | j = 1, \ldots, L\} \\
&= \{h_k^{LM} | j = 0, \ldots, L\}, \\
n_{k,0}^{LM} &= x_k^{LM} \quad \text{(token representation; GloVe)} \\
n_{k,j}^{LM} &= \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \quad \text{(LSTM state)}
\end{align*}
\]

\[
\text{ELMo}_{k}^{\text{task}} = E(R_k; \Theta^{\text{task}}) = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{\text{task}} n_{k,j}^{LM}
\]

\(\gamma^{\text{task}}\): scale (hyper-parameter)

\(s_j^{\text{task}}\): weight (learned)
ELMo for MRC

- ELMo as a word embedding
Transformer

- Parallel self-attention
  - Looks at self, and determines where to focus

\[ \text{Attention}(Q, K, V) = \text{softmax}\left( \frac{QK^T}{\sqrt{d_k}} \right)V \]

Q,K,V – vectors for every word, output attention summed up; head means different Q,K,V vector with different weights

Vaswani, Ashish, et al. "Attention is all you need." \textit{NIPS 2017}
BERT: Bidirectional Encoder Representations from Transformers

- Training 1. Masked words prediction
  - 15% of words are [MASK]ed

*GELU: Gaussian error linear unit
• Training 2. Next sentence prediction
  • To understand texts more than a sentence

Input $= [\text{CLS}]$ the man went to [MASK] store [SEP]
  he bought a gallon [MASK] milk [SEP]
Label $= \text{IsNext}$

Input $= [\text{CLS}]$ the man [MASK] to the store [SEP]
  penguin [MASK] are flight ##less birds [SEP]
Label $= \text{NotNext}$
BERT: Bidirectional Encoder Representations from Transformers

- BERT as universal pre-trained model for NLP
  - BERT requires minimal additional layers and fine-tuning

*GLUE benchmark task

GPT (Generative Pre-Training)

In pre-training, optimize $L_1(u)$
- $u$: Unlabeled dataset
- $\Theta$: Model parameters

In fine-tuning, optimize $L_3(c)$
- $c$: Labeled dataset
- $\lambda$: Hyper-parameter weight

*$\text{GPT-2/3: zero/few-shot learning}$
Sequence labeling

• Sequence labeling is the task of assigning a categorical label to each member of an observed sequences

• Examples of sequence labeling
  • **Part-of-speech tagging** labels each word with a grammatical category
    • e.g. The | trees | are | ... → DT (determiner) | NNS (plural noun) | VBP (plural verb)
  • **Named entity recognition** locates and classifies named entity in text. It can be tackled by labeling each word with a named entity category
    • e.g Barack | Obama | said | ... → B-PERSON | I-PERSON | O | ...
Bi-directional LSTM-CNNs-CRF

- **Bi-directional LSTMs** encode word embeddings and character representations

- **Conditional random fields** compute the distribution of output sequence
  - Viterbi algorithm is applied during training and decoding
  - The objective is the negative log-likelihood of the output sequence distribution

\[
p(y | z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in Y(z)} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)}
\]

- **z**: input sequence
- **y**: output sequence
- **Y(z)**: a set of all possible output sequences when given the input sequence z

[Ma 2016]
S2S NMT via fixed-length representations

- Encoder RNN compresses input sequence into a fixed-length representation
- Decoder RNN produces output sequence from the representation
  - Each produced output token is fed into the next RNN’s input

[Sutskever 2014]
S2S NMT with attention mechanism

- It’s hard to encode all the information of an input sequence into a fixed-length representation.
- We can focus important parts of input sequences for each decoding step by attention mechanism.

[Bahdanau 2015]
Dependency parsing

• Dependency parsing is the task of extracting dependencies between head and dependent words from a sentence.

• A dependency is the arrow from a head to a dependent with a grammatical type called relation (e.g., nsubj).

• Dependencies show which words depend on (modify or are arguments of) which other words.

Root → nsubj → He → has → good → control → punct
PRP VBZ JJ NN .

ROOT He has good control .
PRP VBZ JJ NN .
Neural transition-based dependency parsing

- Extract features from Stack and Buffer
  - lc/rc: leftmost/rightmost children
- Classify an action by neural networks
  - The objective is the negative log-likelihood of the action distribution

Feature extraction

Feedforward neural network-based action classifier

[Chen 2014]
Semantic parsing

• Semantic parsing is a task of mapping natural language to program for specific applications

• Example

Semantic parser

Program
(exist (filter all-boxes (λ (x) (and (exist (filter x (λ (y) (and (is-blue y) (is-circle y))))) (exist (filter x (λ (z) (and (is-black z) (is-circle z))))) )))

Natural language
“There is a box with a blue circle, a black circle and other items.”

Knowledge base

[{{"y_loc":81,"size":10,"type":"circle","x_loc ":51,"color":"#0099ff"}, {...}}]

Denotation
True
**Input:** There is a box with a blue circle, a black circle and other items

**Output:**

\[
(\text{exist \ (filter \ all-boxes} \ (\lambda \ (x) \ (\text{and \ (exist \ (filter \ x} \ (\lambda \ (y) \ (\text{and \ (is-blue} \ y)) \ (\text{is-circle} \ y)))))
\]

\[
(\text{exist \ (filter \ x} \ (\lambda \ (z) \ (\text{and \ (is-black} \ z) \ (\text{is-circle} \ z)))))))
\]
Input: Greece held its last Summer Olympics in which year?
Output: ((reverse year.date) (argmax (country greece) index))
code generation by semantic parsing

Abstract Syntax Description Language (ASDL) for Python

\[\text{stmt} \mapsto \text{Expr} (\text{expr} \text{ value})\]
\[\text{expr} \mapsto \text{Call} (\text{expr} \text{ func}, \text{expr}^* \text{ args}, \text{keyword}^* \text{ keywords})\]
\[\text{expr} \mapsto \text{Name} (\text{identifier} \text{ id})\]
\[\text{Expr} \mapsto \text{expr} [\text{value}]\]

\textbf{ASDL for SQL}

\texttt{stmt = Select(agg_op? agg, idx, column_idx,}
\texttt{ cond_expr* conditions)}

\texttt{cond_expr = Condition(cmp_op op, idx, column_idx,}
\texttt{ string value)}

\texttt{agg_op = Max | Min | Count | Sum | Avg}

\texttt{cmp_op = Equal | GreaterThan | LessThan | Other}

\textbf{Yin 2017, Yin 2018, Rabinovich 2017}
Sentiment Analysis

- interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques
Sentiment Analysis

- XLNet based classification

*XLNet = GPT (AR)+BERT(AE): permutation AR (Transformer-XL)
ConvNet for NLP

RNN for NLP - softmax is often only calculated at the last step

CNN for NLP
ConvNet for NLP

- CNN architecture for sentence classification

MRC-QA: SQuAD2.0

- Unanswerable question (negative example)
  - Relevant to the topic
  - Existence of plausible answers

**Article:** Endangered Species Act

**Paragraph:** “...Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.”

**Question 1:** “Which laws faced significant opposition?”
**Plausible Answer:** later laws

**Question 2:** “What was the name of the 1937 treaty?”
**Plausible Answer:** Bald Eagle Protection Act

Figure 1: Two unanswerable questions written by crowdworkers, along with plausible (but incorrect) answers. Relevant keywords are shown in blue.
SQuAD2.0

- BERT – no answer prediction

answerable case

unanswerable case
MRC - HotpotQA

- multi-hop (two-hop) reasoning
- multiple supporting doc
- sentence-level supporting facts


HGN
Grammatical error correction

- Grammatical error correction (CoNLL 2014 shared task, 1312 pairs)

Raw dataset

S Do one who suffered from this disease keep it a secret of infrom their relatives?
A 0 1 |||SVA |||Does |||REQUIRED|||-NONE-|||0
A 3 4 |||Vt |||suffers|||REQUIRED|||-NONE-|||0
A 11 12|||Trans|||or |||REQUIRED|||-NONE-|||0
A 12 13|||Mec |||inform |||REQUIRED|||-NONE-|||0

Implies...

Do one who suffered from this disease keep it a secret of infrom their relatives?

Does one who suffers from this disease keep it a secret or inform their relatives?
Grammar error correction

- Multi-layer transformer (Zhao et al., NAACL 2019)
  - Transformer + copy architecture

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F$_{0.5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAI 2018</td>
<td>0.6813</td>
<td>0.2345</td>
<td>0.4933</td>
</tr>
<tr>
<td>NAACL 2019</td>
<td>0.6848</td>
<td>0.3310</td>
<td>0.5642</td>
</tr>
</tbody>
</table>
End-to-End Task-oriented dialogue system

• The system interacts with user to help user achieve his goal
  • e.g. Restaurant reservation, hotel reservation...

• Task-oriented dialogue system has some sub-tasks
  • e.g. Automatic speech recognition, spoken language understanding, dialogue state tracking, natural language generation, text-to-speech

• MultiWOZ(Wizard of OZ) dataset
  • Multi-domain dataset for task-oriented dialogue system
  • 2018 ACL @ Cambridge Univ. [https://www.aclweb.org/anthology/D18-1547/]
  • Consists of 13 acts, 7 domains, 25 slots and 69 domain-slot pairs
  • Total number of values in all slots is 4510
  • Number of dialogue is 8,438
  • The average number of turns per dialogue is 13.68

• DAMD(Domain Aware Multi-Decoder)
  • State-Of-The-Art in task-oriented dialogue system
  • 2020 AAAI @ Tsinghua Univ. [https://arxiv.org/abs/1911.10484]
DST (Dialogue State Tracking)

- The most important subtask of task-oriented dialogue system
- Dialogue state generally consists of slot-value pairs
  - Slot means general class like food, area / value means specific value of slot
- In multi-domain, dialogue state consists of domain-slot-value triplets
  - ex) (restaurant-pricerange-cheap), (attraction-type-museum)...
- Infer dialogue state each turn of dialogues between user and system
  - There are ground truth dialogue states every turn
- Inferred state is used to generate next system action and response

TRADE (TRAnsferable Dialogue statE generator)
- 2019 ACL @ Hong Kong Univ. & Salesforce
  [https://www.aclweb.org/anthology/P19-1078]

DS-DST (Dual Strategy for DST)
- State-Of-The-Art in DST
- 2019 Arxiv @ Illinois Univ. & Salesforce
DAMD (Domain Aware Multi-Decoder)

User utterance → Delexicalization → GRU encoder → Attention → User context → Attention → User context → Concat → GRU decoder → Softmax → New belief

System Response → GRU encoder → Attention → Belief context → Attention → Response context → Concat → GRU decoder → Softmax → New action

Action → GRU encoder → Attention → Action context → Concat → GRU decoder → Softmax → New response
End-to-end Neural TTS

• High quality Neural Speech Synthesis
  – Attentive Acoustic Modeling and Parallel WaveNet have been developed to provide high quality & fast voice generation

• Controllable & Expressive Speech Synthesis
  – To provide natural voice expression for Robot interaction, voice should be controlled in terms of prosody, style, and speaker depending on emotion and situation. Eventually, it only requires small amount of voice DB.